A Component Model of Spatial Locality

Xiaoming Gu
Intel China Research Center
Beijing, China

Joint work with Ian Christoper,
Tongxin Bai, Chengliang Zhang
and Chen Ding
Memory Wall and Locality

• Memory wall

• Locality
  • temporal locality: enable cache reuse
  • spatial locality: implicit data prefetching and better cache utilization
Spatial Locality

• Three types
  • intra-block spatial locality ★
  • inter-block spatial locality
  • adjacent-block spatial locality ★
Reuse Distance

- Reuse distance
  - the volume of data referenced between this and the previous access

- Reuse signature
  - the distribution of all finite reuse distances

<table>
<thead>
<tr>
<th>access trace</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>A</th>
<th>A</th>
<th>C</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>rd</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Reuse Distance (in log-2 scale)
Basic Idea

• Spatial locality

• Reduction on reuse distances when measuring block size doubled
Reduction in Reuse Distance

- Two cases of reductions
  - no intercept: reduced by half at most
  - intercept: may reduce it to constant
Contiguous Access

- An array with 8 elements
  - access 0, 1, 2, 3, 4, 5, 6, 7 repeatedly
- Rd measured with single-element block
  - all are 7
- Rd measured with 2-element block
  - half reduced to 0 (constant)
  - half reduced to 3 (dependent on array size)
Measuring Spatial Reuse

• Measure the change of every reuse distance
  • running two reuse distance analyzers simultaneously

• Criteria for effective spatial reuse
  • machine independent (reducing factor \( \geq 8 \))
  • machine dependent (L1&L2 cache sizes)
Quantitative Scoring

- Spatial-locality quality score (SLQ)
- portion of effective spatial reuse for each range of reuse distances, normalized to the best case (contiguous access)
  - 0 means no spatial locality and 1 means best spatial locality
Locality Components

- Program level
- functions or loops
- Behavior level
  - adjacent reuse distance ranges in length, which forms a peak
    - similar in temporal locality but maybe very different in spatial locality
Locality Components

- Three metrics
  - total size
  - temporal locality: average reuse distance
  - spatial locality: average SLQ
- More findings than a single global result
Locality Profiling Tools

- Full trace analysis
  - dynamic binary instrumentation based on Valgrind
  - static source instrumentation based on GCC
- Sampling analysis
  - reservoir sampling (based on SLO [Beyls & D’Hollander])
All Benchmark Results

64-byte block spatial locality score vs. log2 reuse distance (temporal locality)

Points represent different benchmarks: swim.opt-c1, eqauke-c2, swim-c2, mcf-c2, milc, parser, art-r1, art-r2, twolf, gzip-r5, gzip-r1, bzip, gzip-c1, vpr, gzip-r3, gzip-r2, gzip-r4, and crafty.
Result for Array Regrouping on Swim

component 1 improved significantly

component 2 has perfect locality in both versions
A User Study

- Locality tuning for a 2200-line machine translation application
  - using GCC-based tool with calling context tree
  - worst spatial locality occurs in a 10-line library function
  - 7% improvement after 6-line change
  - acceptable loss of precision in the translation
Spatial Locality Ranking

- A sampling-based tool extended from SLO
- Worst spatial locality problems identified for mcf
Summary

• A new model for spatial locality
  • based on reuse distance
  • models intra-block and adjacent-block spatial locality both
  • component-based, on both behavior and program levels
  • promising as part of performance tools
Q & A